

明 細 書

誤り検出装置、及び誤り検出方法

5 技術分野

本発明は、誤り検出装置及び誤り検出方法に関し、特に誤り訂正と誤り検出を同時に実行する誤り検出装置及び誤り検出方法に関する。

背景技術

- 10 デジタルデータの記録／再生を行うシステムでは、再生時もしくは記録時にデータ中に誤りが発生することがあることから、その誤りを訂正し、行った誤り訂正が正しいものかどうかを判定する誤り検出を行う必要がある。

- 例えば、光記録媒体であるDVDの記録／再生において、誤り訂正はECC (Error Correcting Code)、誤り検出はEDC (Error Detecting Code) と呼ばれる方法が用いられる。
- 15

- 前記ECCは、誤り訂正単位ブロック（以下、「ECCブロック」と称す。）単位で処理が実行され、該ECCブロックは、第11図に示すように16セクタより構成されている。そして、前記EDCは、前記ECCブロック中のセクタ単位でその処理が実行されるものであり、1セクタは第12図に示すような構成を有する。
- 20

以下、第11図～第14図を用いて、ECC復号の概要を、光記録媒体であるDVDに記録されているデータを誤り訂正する場合を例に挙げて説明する。

- まず、DVDに記録されているデータ符号を誤り訂正する場合、DVDから読み出したECC符号化されたデータ符号を復号し、そのデータに対して、第11
- 25 図に示すC1方向もしくはC2方向に誤り訂正を行う。このとき、ECC復号したデータから位置多項式及び数値多項式を生成し、それらの根を求めることにより、誤りデータ位置及び誤りデータ数値を算出する。

そして、1ECCブロック中の符号列に対して、前記C1方向あるいはC2方向に誤り訂正を行った際、そのデータ中に誤り訂正能力を超える誤りが存在した

場合は、その誤り訂正能力を超えるデータ符号列を第13図に示すように訂正不能符号列とし、この訂正不能符号列に関する情報を消失位置情報として記憶しておく。

前述のようにして、1 ECCブロック中の全ての符号列に対して、第11図に示すC1方向もしくはC2方向に誤り訂正が完了した後、今度は、前回と異なる方向（C2方向もしくはC1方向）に、前記消失位置情報を用いて、前記1 ECCブロック中の符号列に対して誤り訂正を行う。

前記消失位置情報を用いて誤り訂正を行う場合は、予め誤りデータ位置がわかっているため、前記多項式を生成する際に、数値多項式のみを求めれば良いことになる。その結果、誤り訂正能力を向上させることができる。

例えば、最初に、第13図に示すように、1 ECCブロック中の全ての符号列に対してC1方向に誤り訂正を行ったとき、1 ECCブロック中の50、90、130、200行目が訂正不能符号列であったとする。この後、第14図に示すように、前回と異なる方向であるC2方向に誤り訂正を行うのだが、この時前記訂正不能符号列を示す消失位置情報をもとに、1 ECCブロック中の50、90、130、200 Byte目を消失位置情報と指定し、前記C2方向の誤り訂正能力を向上させることができる。この方法により、伝送系で発生したバースト誤りに対して訂正能力を落とすことなく誤り訂正が可能となる。

そして、データ符号列に対して誤り訂正及び誤り検出を行う場合、従来においては、前述のECC処理が1 ECCブロック分終了した後に、該1 ECCブロックのセクタ単位でEDC演算を行っていた。

以下、第14図及び第15図を用いて、EDC演算の概要について説明する。第15図はEDC演算回路の構成例を示す図であり、ここでは、4 Byte入力のEDC演算回路を示す。

まず、EDCの演算式は下記のようになる。

(数1)

$$EDC(x) = \sum_{i=31}^0 b_i x^i = l(x) \bmod \{g(x)\}$$

$$l(x) = \sum_{i=16511}^{32} b_i x^i \quad g(x) = x^{32} + x^{31} + x^4 + 1$$

前記EDC演算式は、第15図のEDC演算回路に示されるように、32ビットのシフトレジスタの、0ビット目、4ビット目、31ビット目への入力にEXOR演算回路が配置され、該EXOR演算回路にて、それぞれ、4Byteの最上位からのビット入力と31ビット目、3ビット目と31ビット目、及び30ビット目と31ビット目のEXORが演算される。

以上を示したEDC演算式、あるいは第15図のEDC演算回路から理解できるように、EDC演算はデータの記録方向（第11図に示されるC1方向）に沿った線形演算である。

そして、前記第12図に示すような構成を有する1セクタを、データの記録方向、つまりID領域、IEC領域、RSV領域、ユーザデータ領域、EDC領域の順序で、それぞれ4Byte毎に、第15図に示すEDC演算回路に入力すると、最終の4ByteであるEDC領域を入力した後の32ビットのシフトレジスタ値がEDC演算結果となる。

このように、従来では一般的に、誤り訂正（ECC）を行った後に、誤り検出（EDC）を行う方法が用いられているが、この方法では、誤り訂正を行うための一単位であるECCブロックをバッファから読み出して誤り訂正を行った後、誤り検出を行うために、再びECCブロックを前記バッファから読み出さなければならないため、メモリバッファのバンド幅の消費、処理時間の肥大化という問題が生じる。

このようなメモリバッファのバンド幅の消費、あるいは処理時間の肥大化の問題を解決する方法としては、ECCとEDCとの両方の処理を、バッファよりECCブロックを一度読み出すのみで、同時に処理することが考えられる。

しかし、前述したように、EDC演算処理はデータの記録方向（第11図のC1方向）に沿った線形演算であるため、誤り訂正回路にて、第13図に示すように入力されるデータの並びに連続性がある、C1方向に誤り訂正を行う場合は、該C1方向のECCと、前記EDC演算とを同時に処理することが可能であるが、第14図に示すように入力されるデータの並びに連続性がない、C2方向に誤り訂正を行う場合は、該C2方向のECCと、前記EDC演算とを同時処理するのは難度が高く、このような同時処理を半導体の実装するのは困難、という課題が

ある。

本発明は前記課題を解決するためになされたものであり、ECCブロックの誤り検出を行う対象符号列が連続性のない並び（第11図のC2方向）で入力された際にも、誤り訂正と誤り検出を同時に行える誤り検出装置及び誤り検出方法を
5 提供することを目的とするものである。

発明の開示

本発明の誤り検出装置は、それぞれマトリクス状のデータからなる複数のセクタにより構成される対象符号列に対してシンδροーム演算を行うシンδροーム演算器を有し、該対象符号列に対して誤り訂正回路による誤り訂正処理を行うと同時に、該対象符号列に対して前記セクタ単位で誤り検出を行う誤り検出装置であって、前記対象符号列の誤り検出符号を演算する誤り検出符号演算回路と、前記対象符号列が連続性のない並びで入力された際に、該符号列の並びを連続させるようにデータをスキップさせて入力データ間の連続性を補正するスキップ演算を行う誤り検出符号スキップ演算回路と、前記シンδροーム演算と同じ時点で行われる、前記誤り検出符号演算回路あるいは誤り検出符号スキップ演算回路による第1の誤り検出符号演算処理を制御する第1の誤り検出制御回路と、前記誤り訂正処理後に、該誤り訂正処理より得られた誤りデータ位置及び誤りデータ数値を元に、該誤りデータ位置が示すデータに対してのみ行う第2の誤り検出符号演算
10 処理を制御すると共に、該第2の誤り検出符号演算処理による演算結果を元に、前記第1の誤り検出符号演算処理の演算結果を更新する更新処理を制御する第2の誤り検出制御回路と、前記誤り検出符号演算回路、及び前記誤り検出符号スキップ演算回路による演算結果を保持するメモリと、を備えるものである。

これにより、ECCブロックの誤り検出を行う対象符号列が連続性のない並び
25 で入力された際にも、誤り訂正と誤り検出を同時に行える。

さらに、本発明の誤り検出装置は、前記誤り検出符号スキップ演算回路は、前回までに入力された対象符号列の誤り検出符号を入力とし、該誤り検出符号スキップ演算回路に予め設定されたスキップ演算を行うものである。

これにより、簡単な処理でスキップ演算が行え、誤り訂正と誤り検出の同時実

行を、半導体回路に実装可能にする。

さらに、本発明の誤り検出装置は、前記メモリは、前記対象符号列が連続性のある並びで入力された際は、前記誤り検出符号演算回路、及び前記誤り検出符号スキップ演算回路によるセクタ単位の演算結果を保持し、前記対象符号列が連続性のない並びで入力された際は、前記誤り検出符号演算回路、及び前記誤り検出符号スキップ演算回路によるセクタ毎の途中演算結果を保持する第1のメモリと、前記第1のメモリから送信される演算結果を、前記各セクタ毎に保持する第2のメモリと、を備えるものである。

これにより、ECCブロックの誤り検出を行う対象符号列が連続性のない並びで入力された際にも、セクタ毎のEDC演算結果を得ることができる。

さらに、本発明の誤り検出装置は、前記第1のメモリは、前記誤り検出符号演算回路、及び前記誤り検出符号スキップ演算回路で実行される、前記第1の誤り検出符号演算処理による演算結果を保持するメモリと、前記第2の誤り検出符号演算処理による、前記第1の誤り検出符号演算処理の演算結果を更新する差分演算結果を保持するメモリとを含むものである。

これにより、簡単な処理で正しいEDC演算結果が得ることが可能となり、誤り訂正と誤り検出の同時実行を、半導体回路に実装可能にする。

さらに、本発明の誤り検出装置は、前記誤り検出符号スキップ演算回路は、前記対象符号列が連続性のない並びで入力されたとき、該対象符号列のうち前記セクタの非最終行においては、一定のバイト数をスキップさせるスキップ演算を行い、前記対象符号列のうち前記セクタの最終行においては、該データが存在する列位置に応じたバイト数をスキップさせる個別のスキップ演算を行うものである。

これにより、前記対象符号列が連続性のない並びで入力されたときにも、EDC演算処理を行うことができ、誤り訂正と誤り検出の同時実行を、半導体回路に実装可能にする。

さらに、本発明の誤り検出装置は、前記対象符号列の該スクランブル成分を一括除去するスクランブル除去部を含むものである。

これにより、処理資源を最小化できる。

さらに、本発明の誤り検出装置は、前記スクランブル除去部は、前記対象符号

列のスクランブル成分を除去するためのデータを保持するテーブルを有するものである。

これにより、前記対象符号列のスクランブル成分を一括除去することが可能となる。

- 5 また、本発明の誤り検出方法は、それぞれマトリクス状のデータからなる複数のセクタにより構成される、連続性のない並びで入力された対象符号列に対して誤り訂正単位ブロック単位で誤り訂正を行うと同時に、該連続性のない並びで入力される対象符号列に対して前記セクタ単位で誤り検出を行う誤り検出方法であって、前記対象符号列に対してシンドローム演算を行うシンドローム演算ステップと、前記シンドローム演算ステップと同時に行われる、該連続性のない並びで入力される対象符号列に対して誤り検出符号演算を行う第1の誤り検出符号演算ステップと、前記シンドローム演算ステップにおいて得られたシンドロームに基づいて、前記対象符号列の誤りデータ位置及び誤りデータ数値を計算して誤り訂正を行う誤り訂正ステップと、前記誤り訂正ステップにおいて得られた前記誤りデータ位置及び誤りデータ数値を元に、前記対象符号列のうちの前記誤りデータ位置に対してのみ再度誤り検出符号演算を行う第2の誤り検出符号演算ステップと、前記第2の誤り検出符号演算ステップによる演算結果を用いて、前記第1の誤り検出符号演算ステップによる演算結果を更新する更新ステップと、を含み、前記第1の誤り検出符号演算ステップ及び第2の誤り検出演算ステップは、前記対象符号列の誤り検出符号を演算する誤り検出符号演算ステップと、前記連続性のない並びで入力された対象符号列の並びを連続させるようにデータをスキップさせてデータ間の連続性を補正するスキップ演算を行う誤り検出符号スキップ演算ステップと、を含むものである。
- 10 プと、前記シンドローム演算ステップと同時に行われる、該連続性のない並びで入力される対象符号列に対して誤り検出符号演算を行う第1の誤り検出符号演算ステップと、前記シンドローム演算ステップにおいて得られたシンドロームに基づいて、前記対象符号列の誤りデータ位置及び誤りデータ数値を計算して誤り訂正を行う誤り訂正ステップと、前記誤り訂正ステップにおいて得られた前記誤りデータ位置及び誤りデータ数値を元に、前記対象符号列のうちの前記誤りデータ位置に対してのみ再度誤り検出符号演算を行う第2の誤り検出符号演算ステップと、前記第2の誤り検出符号演算ステップによる演算結果を用いて、前記第1の誤り検出符号演算ステップによる演算結果を更新する更新ステップと、を含み、前記第1の誤り検出符号演算ステップ及び第2の誤り検出演算ステップは、前記対象符号列の誤り検出符号を演算する誤り検出符号演算ステップと、前記連続性のない並びで入力された対象符号列の並びを連続させるようにデータをスキップさせてデータ間の連続性を補正するスキップ演算を行う誤り検出符号スキップ演算ステップと、を含むものである。
- 15 データ位置及び誤りデータ数値を元に、前記対象符号列のうちの前記誤りデータ位置に対してのみ再度誤り検出符号演算を行う第2の誤り検出符号演算ステップと、前記第2の誤り検出符号演算ステップによる演算結果を用いて、前記第1の誤り検出符号演算ステップによる演算結果を更新する更新ステップと、を含み、前記第1の誤り検出符号演算ステップ及び第2の誤り検出演算ステップは、前記対象符号列の誤り検出符号を演算する誤り検出符号演算ステップと、前記連続性のない並びで入力された対象符号列の並びを連続させるようにデータをスキップさせてデータ間の連続性を補正するスキップ演算を行う誤り検出符号スキップ演算ステップと、を含むものである。
- 20 対象符号列の誤り検出符号を演算する誤り検出符号演算ステップと、前記連続性のない並びで入力された対象符号列の並びを連続させるようにデータをスキップさせてデータ間の連続性を補正するスキップ演算を行う誤り検出符号スキップ演算ステップと、を含むものである。

- 25 これにより、ECCブロックの誤り検出を行う対象符号列が連続性のない並びで入力された際にも、誤り訂正と誤り検出を同時に行える。

さらに、本発明の誤り検出方法は、前記誤り検出符号スキップ演算ステップは、前回までに入力された対象符号列の誤り検出符号を入力とし、予め設定されたスキップ演算を行うものである。

これにより、簡単な処理でスキップ演算が行え、入力されるデータの並びに連

続性がない場合でも、誤り訂正と誤り検出との同時実行を、容易に実現することができる。

さらに、本発明の誤り検出方法は、前記誤り検出符号スキップ演算ステップは、前記対象符号列のうち前記セクタの非最終行においては、一定のバイト数をスキップさせるスキップ演算を行い、前記対象符号列のうち前記セクタの最終行においては、該データが存在する列位置に応じたバイト数をスキップさせる個別のスキップ演算を行うものである。

これにより、前記対象符号列が連続性のない並びで入力されたときにも、EDC演算処理を行うことができ、誤り訂正と誤り検出の同時実行を、半導体回路に実装可能にする。

さらに、本発明の誤り検出方法は、前記個別のスキップ演算は、前記データが存在する列位置のうち、特定の列位置で実行されるステップ演算の演算結果を複数回利用して行うものである。

これにより、セクタ内の誤り検出演算処理単位列に対応した、全てのスキップ演算処理を用意することなく、別個に存在するスキップ演算を複数回再利用することができ、この結果、処理資源の消費を抑えることが可能となる。

さらに、本発明の誤り検出方法は、前記更新ステップと同時に行われる、前記対象符号列に含まれるスクランブル成分を除去するスクランブル除去ステップを含むものである。

これにより、ECC処理及びEDC処理後のデータから、スクランブル成分を除去できる。

さらに、本発明の誤り検出方法は、前記スクランブル除去ステップは、1セクタの全データが入力された後に、該セクタの全データのスクランブル成分を一度に除去するものである。

これにより、スクランブル成分の除去処理に必要な処理資源を削減できる。

さらに、本発明の誤り検出方法は、前記スクランブル除去ステップは、前記対象符号列のスクランブル成分を除去するためのデータを保持するテーブルを用いて行われるものである。

これにより、ECC処理及びEDC処理後のデータから、全データのスクラン

ブル成分を一度に除去することができる。

図面の簡単な説明

第 1 図は、本発明の実施の形態 1 における、ECC 処理と EDC 処理の同時実
5 行した際のパイプライン処理を示す図である。

第 2 図は、本発明の実施の形態 1 における、誤り検出回路の構成を示す図である。

第 3 図は、本発明の実施の形態 1 における、C2 方向訂正時に第 0 セクタにおいてデータ入力単位を 4 Byte としたときの誤り訂正回路、及び誤り検出回路
10 に対するデータ入力順序を示す図である。

第 4 図は、本発明の実施の形態 1 における誤り検出回路において、C2 方向訂正時の第 1 の EDC 演算処理の一連の流れを示すフローチャート図である。

第 5 図は、本発明の実施の形態 1 における誤り検出回路の C2 方向訂正時における第 1 の EDC 演算処理のデータの流れを示す図である。

15 第 6 図は、本発明の実施の形態 1 における誤り検出回路の、C2 方向訂正時における第 2 の EDC 演算処理のデータの流れを示す図である。

第 7 図は、本発明の実施の形態 1 における誤り検出回路の、誤り検出符号スキップ演算回路において、データを 168 Byte スキップさせる EDC スキップ演算処理を示す図である。

20 第 8 図は、本発明の実施の形態 1 における誤り検出回路の、誤り検出符号スキップ演算回路において、データを 168 Byte スキップさせる EDC スキップ演算処理の分割処理を示す図である。

第 9 図は、本発明の実施の形態 1 における、対象符号列が入力される度にスクランブル成分を除去する場合の、誤り検出回路の構成図である。

25 第 10 図は、本発明の実施の形態 1 における、データ符号列が入力される度にスクランブル成分を付加する場合の、誤り検出回路の構成図である。

第 11 図は、DVD に記録されたデータを誤り訂正単位ブロック（ECC ブロック）に分けたときの 1 ECC ブロック構成図である。

第 12 図は、ECC ブロック上のセクタの構成を示す図である。

第13図は、ECCブロックの第11図に示すC1方向についての誤り訂正実施例を示す図である。

第14図は、ECCブロックの第11図に示すC2方向についての誤り訂正実施例を示す図である。

5 第15図は、4Byte入力のEDC演算回路を示す図である。

発明を実施するための最良の形態

実施の形態1.

以下、本実施の形態1における、誤り訂正と誤り検出を同時処理する概要を、
10 第1図を用いて説明する。なお、本実施の形態1においては、光記録媒体であるDVDに記録されているデータを誤り訂正（ECC）及び誤り検出（EDC）する場合を例に挙げて説明する。また、誤り訂正を行う一単位である誤り訂正単位ブロック（以下、「ECCブロック」と称す。）は、第11図に示すように16のセクタより構成され、1セクタは、第12図に示す構成を有するものとする。

15 第1図は、ECC処理とEDC処理の同時実行におけるパイプライン処理を説明する図である。

ECC処理とEDC処理とを同時実行する場合、まず、ステップS10において、DVDから、誤り訂正及び誤り検出を行うべき対象データ符号列を、第11図に示されるように1ECCブロックのC1方向あるいはC2方向に入力して、
20 ECC処理のシンドローム演算を行う。このシンドローム演算結果は、その値が“0”のとき誤りが存在しないことを示し、“0”でないとき誤りが存在することを示す。

そして、前記ステップS10と同じパイプライン処理ステージであるステップS20において、前記ステップS10と同じ順番で入力されるデータ符号列に対して、第1のEDC演算を実行し、セクタ単位毎に第1の誤り検出符号演算結果
25 を得る。この第1の誤り検出符号演算結果は、入力されたデータ符号列に誤りが含まれていない場合は正常な値となり、誤りが含まれている場合は不正な値となる。

そして、前記ステップS10により得られるシンドローム演算結果が“0”で、

入力されたデータ符号列に誤りが存在しないと判定された場合は、ステップS 30の誤り訂正処理は行われずに、一連の処理を終了する。そしてこの場合、前記ステップS 20において得られた第1の誤り検出符号演算結果が、当該誤り検出装置より得られる正しいEDC演算結果として出力される。

- 5 一方、ステップS 10により得られるシンドローム演算結果が“0”とならず、入力されたデータ符号列に誤りが存在すると判定された場合は、該データ符号列に対して誤り訂正処理が行われ、誤りデータ位置及び誤りデータ数値を得る（ステップS 30）。

- 10 そして、ステップS 40において、前記ステップS 30で得られた誤りデータ位置、及び誤りデータ数値を元にして、DVDから入力されるデータ符号列のうち、該得られた誤り位置が示すデータに対してのみ、第2のEDC演算処理を実行して、第2の誤り検出符号演算結果を得る。このときに得られる第2の誤り検出符号演算結果は、当該誤り検出装置より得られる正しいEDC演算結果と、前記ステップS 20において得られた前記第1の誤り検出符号演算結果との差分情報である。

15 よって、ステップS 50において、前記ステップS 40で得られた第2の誤り検出符号演算結果を用いて、前記ステップS 20において得られた第1の誤り検出符号演算結果を更新し、その更新後の誤り検出符号演算結果が、当該誤り検出回路より得られる正しいEDC演算結果として出力される。

- 20 さらに、前記ステップS 50と同じパイプライン処理ステージであるステップS 60で、スクランブル成分を除去するためのデータを保持するテーブルを用いて、入力されるデータ符号列のスクランブル成分を一括で除去する処理が行われる。

- 25 以上のような一連の処理を行うことにより、ECC処理と、EDC処理、及び入力されるデータ符号列のスクランブル成分除去処理を、ECCブロックをバッファから1度だけ読み出すことによって、実行することが可能となる。

なお、第1図では、データ符号列のスクランブル成分を、EDC演算処理後に一括除去するものとして説明したが、前記スクランブル成分をEDC演算処理後に一括除去するのではなく、ステップS 20のEDC演算処理の際、前記データ

符号列が入力される度に除去しながら演算処理するようにしてもよい。

また、前述の説明では、DVDに記録されているデータを再生する場合を例に挙げ、入力データ符号列にスクランブル成分が含まれている場合を想定しているが、入力データ符号列にスクランブル成分が含まれていない場合もある。例えば、DVDにデータを記録する場合は、入力データにはスクランブル成分は含まれていない。従ってこの場合、ステップS10のシンドローム演算処理の際、前記データ符号列が入力される度にスクランブル成分を付加しながら演算処理する。なお、スクランブル成分をデータ符号列に付加する場合は、ステップS60に示すスクランブル除去のように一括して行うことはできない。

- 10 次に、ECC処理とEDC処理の同時処理が実行可能な、本実施の形態1における誤り検出回路の構成について、第2図を用いて説明する。なお、ここでは、第1図を用いて説明したように、入力されるデータ符号列にスクランブル成分が含まれ、該スクランブル成分を一括除去する場合の構成について説明する。

第2図は、本実施の形態1にかかる誤り検出回路の構成を示す図である。

- 15 本実施の形態1に係る誤り検出回路10は、前記データ符号列に対してシンドローム演算を行うシンドローム演算器21を有する誤り訂正回路20がECC処理を実施するのと同時に、EDC処理を実施可能なものであって、第2図に示すように、誤り検出符号演算回路111と誤り検出符号スキップ演算回路112とからなる演算回路11と、第1の誤り検出符号演算結果保持部131と第2の誤り検出符号演算結果保持部132とからなる第1のメモリ13と、第2のメモリ12と、第1の誤り検出制御回路141と第2の誤り検出制御回路142とからなる制御回路14とで構成される。そして、スクランブル除去部15は、データ符号列のスクランブル成分を除去するものである。
- 20

- 以下、各回路について説明すると、前記制御回路14は、当該誤り検出回路10に入力されるデータ符号列に対して誤り検出を行う際に、当該誤り検出回路10を制御するものであり、前記第1の誤り検出制御回路141は、前記誤り訂正回路20内のシンドローム演算器21における前記データ符号列に対するシンドローム演算と同時に行われる第1のEDC演算処理（第1図のステップS20）を制御し、前記第2の誤り検出制御回路142は、前記誤り訂正回路20による
- 25

誤り訂正処理後の第2のEDC演算処理、及び前記第1のEDC演算結果の更新処理（第1図のステップS40,S50）を制御する。

そして、前記演算回路11は、前記制御回路14の制御の下、入力される前記データ符号列に対して誤り検出演算を行うものであり、前記誤り検出符号演算回路111は、前記データ符号列が第11図のC1方向、つまり連続性のある並びで入力された際に、該データ符号列の誤り検出符号を任意のデータ毎（ここでは4バイト毎）に演算するものであり、前記誤り検出符号スキップ演算回路112は、前記データ符号列が第11図のC2方向、つまり連続性のない並びで入力された場合に、該入力データに連続性をもたせるように、データをスキップさせて入力データ間の連続性を補正するスキップ演算を行う。

そして、前記第1のメモリ13は、前記演算回路11において得られた途中演算結果を保持していくものであり、前記第1の誤り検出符号演算結果保持部131は、前記第1の誤り検出制御回路141の制御の下で行われる第1のEDC演算（第1図のステップS20）で得られた演算結果を保持するものであり、前記第2の誤り検出符号演算結果保持部132は、前記第2の誤り検出制御回路142の制御の下で行われる第2のEDC演算（第1図のステップS40）で得られた演算結果、つまり前記第1のEDC演算処理による演算結果を更新するための差分情報を保持するものである。

そして、前記第2のメモリ12は、前記第1のメモリ13から出力された演算結果をセクタ毎に保持するものであり、前記第2のEDC演算処理後に、前記第1のEDC演算結果が前記第2のEDC演算結果により更新され、該第2のメモリ12には、正しいEDC演算結果が保持される。

以下、ECC処理とEDC処理の同時実行処理動作を説明する。

まず、誤り訂正回路20で、対象データ符号列が第11図のC1方向に訂正処理される場合、誤り訂正回路20及び誤り検出回路10に対して、DVDから、誤り訂正及び誤り検出を行うべきデータ符号列が、第11図のC1方向に沿って、連続性のある並びで、1ECCブロック分入力される。データ符号列が入力されると、誤り訂正回路20では、シンδροーム演算器21においてECC処理のシンδροーム演算が行われる（第1図のステップS10）と同時に、誤り検出回路

10では、演算回路11の誤り検出符号演算回路111で、第1の誤り検出制御回路141の制御の下、第1のEDC演算が行われ(第1図のステップS20)、セクタ単位毎に第1の誤り検出符号演算結果が得られる。この第1の誤り検出符号演算結果は、第1のメモリ13の第1の誤り検出符号演算結果保持部131に保持される。

一方、誤り訂正回路20で、対象データ符号列が第11図のC2方向に訂正処理される場合は、誤り訂正回路20及び誤り検出回路10に対して、DVDから、誤り訂正及び誤り検出を行うべきデータ符号列が、第11図のC2方向に沿って、連続性のない並びで、1ECCブロック分入力される。

10 具体的には、誤り訂正回路20及び誤り検出回路10に対して入力されるデータ符号列は、第3図に示すようになる。第3図は、本実施の形態1における、1ECCブロックの第0セクタのデータ符号列に対してC2方向に誤り訂正処理される際の、誤り訂正回路及び誤り検出回路に対するデータ符号列のデータ入力順序及びEDC演算順序を示す図である。

15 このような連続性のない並びでデータ符号列が入力されると、誤り訂正回路20内のシンドローム演算器21にて、シンドローム演算が実行されると同時に(第1図のステップS10)、誤り検出回路10内の演算回路11の誤り検出符号スキップ演算回路112にて、第1の誤り検出制御回路141の制御の下、第1のEDC演算処理が行われる。

20 以下、誤り訂正回路にてC2方向訂正時に同時にEDC演算処理を行う場合について、第4図及び第5図を参照しながら説明する。

第4図は、本実施の形態1における誤り検出回路にデータ符号列が連続性のない並びで入力された際の、第1のEDC演算処理を示すフローチャート図、第5図は、第1のEDC演算処理時のデータの流れを示す図である。

25 第4図に示すように、第1のEDC演算処理が開始されると、まず、第1の誤り検出制御回路141内に保持されているセクタ内行数カウンタ“ROW”と、セクタ内列数カウンタ“COL”とを初期化する(ステップS201、S202)。そして、第1のメモリ13内の第1誤り検出符号演算結果保持部131の値(以下、“TMP_EDC_VAL_1”)と称す。)、及び第2のメモリ12の値(以下、

「“EDC_VAL”」と称す。)を初期化する(ステップS203, S204)。

この後、誤り検出回路10及び誤り訂正回路20において、前記データ符号列を4Byte受信する。この受信したデータは、第3図に示す“0”番目のデータである(ステップS205)。

- 5 そして、受信したデータに対して、誤り検出符号演算回路111において、後述する4Byte入力のEDC演算処理を行い(ステップS206)、その演算結果と、“TMP_EDC_VAL_1”のEXORをとった値を、再度“TMP_EDC_VAL_1”に保持する(ステップS207)。

- 10 以上のようにして“0”番目のデータのEDC演算が終了後、次に入力されるデータ符号列は、第3図に示す“1”番目のデータである。この次に入力される“1”番目のデータは、第3図から明らかなように、先に入力された“0”番目のデータとデータの並びに連続性をもたない。

- 15 データの並びに連続性があるか否かは、前記第1の誤り検出制御回路141にて、現在処理中のセクタ内行数カウンタ“ROW”を判定し、該セクタ内行数カウンタ“ROW”が、セクタ内最終行である“11”でなければ、データに連続性がないと判断する(ステップS208)。

- 20 そして、前記第1の誤り検出制御回路141にて、データに連続性が無いと判断されると、“0”番目のデータと“1”番目のデータ間の連続性を補正するために、前記第1の誤り検出制御回路141の制御の下、前記演算回路11内の誤り検出符号スキップ演算回路112によって、168ByteスキップEDCスキップ演算処理を行う(ステップS209)。この処理の詳細については後述する。

そして、この誤り検出符号スキップ演算回路112により得たスキップ演算結果と、前記“TMP_EDC_VAL_1”のEXORをとった値を、再度“TMP_EDC_VAL_1”に保持する(ステップS210)。

- 25 この後、前記第1の誤り訂正検出制御回路141は、セクタ内行数カウンタ“ROW”を1インクリメントし(ステップS211)、次行のEDC演算処理に移行する。

以上の処理を、前記セクタ内行数カウンタ“ROW”がセクタ内最終行である“11”になるまで繰り返す。

ステップS 2 0 8において、前記セクタ内行数カウンタ“ROW”が“1 1”になった場合は、第3図に示す“1 1”番目、もしくは“2 1 9”番目、・・・、“8 5 3 9”番目のデータ入力に対応した4 B y t e入力のEDC演算処理が終了している状態となる。

- 5 そして、前記セクタ内列数カウンタ“COL”が“4 2”になるまで、“COL”の値毎に前記誤り検出符号スキップ演算回路1 1 2において行うEDCスキップ演算処理は異なるため、前記セクタ内行数カウンタ“ROW”が“1 1”になる度に、前記第1の誤り検出制御回路1 4 1により、現在のセクタ内列数カウンタ“COL”の値に対応した個別の処理（第4図のステップS 2 1 3参照）となる
- 10 よう制御される。

- 例えば、ステップS 2 1 3の時点で、現在前記セクタ内列数カウンタ“COL”が“1”、つまり、第3図に示す“2 1 9”番目のデータまでEDC演算処理が終了している状態であれば、該“2 1 9”番目のデータと、第3図に示すセクタの最終データでありEDC領域である“8 7 4 7”番目のデータまでの差分は1 6
- 15 4 B y t eであるため、前記第1の誤り検出制御回路1 4 1は、前記誤り検出符号スキップ演算回路1 1 2において、1 6 4 B y t eスキップさせるEDCスキップ演算処理を行うように制御する。また、前記ステップS 2 1 3に時点で、前記セクタ内列数カウンタ“COL”が“4 1”、つまり第3図に示す“8 5 3 9”番目のデータまでのEDC演算処理が終了している状態であれば、該“8 5 3 9”
- 20 番目のデータと、第3図に示す最終データでありEDC領域である“8 7 4 7”番目のデータまでの差分は4 B y t eであるため、前記第1の誤り検出制御回路1 4 1は、前記誤り検出符号スキップ演算回路1 1 2において、4 B y t eスキップさせるEDCスキップ演算処理を行うように制御する。

- そして、前記セクタ内列数カウンタ“COL”値毎に異なる、前記誤り検出符号スキップ演算回路1 1 2によるスキップ演算結果と、“TMP_EDC_VAL_1”とのEXORをとった値を、再度“TMP_EDC_VAL_1”に保持する（ステップS 2 1 4）。
- 25

この時点で、第1のメモリ1 3内の、第1の誤り検出符号演算結果保持部1 3 1（“TMP_EDC_VAL_1”）には、セクタ内1列分の演算結果が保持された

- こととなるので、第5図に示すように、第1の誤り制御回路141により、この保持された演算値を第2のメモリ12 (“EDC_VAL”) に出力して、該第2のメモリ12内の対応するセクタレジスタに出力して保持する。つまり、“EDC_VAL” と “TMP_EDC_VAL_1” とのEXORをとった値を、再度 “EDC_VAL” に保持する (ステップS215)。例えば、現在第0セクタ内の “COL” が “1”、つまり第3図に示す “219” 番目のデータまでのEDC演算が終了し、“TMP_EDC_VAL_1” には、第3図の “208” 番目～ “219” 番目のデータまでのEXORをとった値が保持されている場合、第5図に示すように、この値を “EDC_VAL” 内の第0セクタレジスタに出力する。
- 10 この後、“ROW” を初期化し (ステップS216)、“COL” を1インクリメントする (ステップS217)。
- 一方、前記ステップS208にて “ROW” が “11” で、前記ステップS212にて “COL” が “42” であった場合には、第3図に示す “8747” 番目のデータ入力に対応した4Byte入力EDC演算処理が終了した状態であるため、“EDC_VAL” と “TP_EDC_VAL_1” とのEXORをとった値を、再度 “EDC_VAL” に保持し (ステップS218)、“ROW” 及び “COL” を初期化し (ステップS219, S220)、処理を終了する。
- 15 以上のステップにより、第3図に示すECCブロック内第0セクタの第1のEDC演算結果が全て求まり、処理を終了する。
- 20 このようにして、前記ステップS20にて、1ECCブロック内の16セクタ全てに対する第1のEDC演算処理が終了した後、前記ステップS10で、前記誤り訂正回路20におけるシンδροーム演算値が “0” でなく、誤りが含まれていると判断された場合は、誤り訂正処理が施される (第1図のステップS30)。なお、前記シンδροーム演算値が “0” の場合は、ここで一連の処理は終了され、前記第1のEDC演算結果が、誤り検出回路10より得られる正しいEDC演算結果として出力される。
- 25

そして、前記誤り訂正回路20におけるシンδροーム演算値が “0” でなかった場合は、前記ステップS30の誤り訂正処理において得られた誤りデータ位置及び誤りデータ数値を用いて、該誤りデータ位置が示す箇所のみ第2のEDC演

算処理が行われる。

この第2のEDC演算処理については、第2の誤り検出制御回路142の制御の下、前述した第1のEDC演算処理と同様、演算回路11内の誤り検出符号演算回路111及び誤り検出符号スキップ演算回路112によって、第2のEDC
5 演算処理が行われる。但し、入力される前記データ符号列のうち、前記ステップS30の誤り訂正回路20による誤り訂正処理において得られた誤りデータ位置が示すデータについてのみ行う。この第2のEDC演算結果は、第6図の、第2のEDC演算処理時のデータの流れに示されるように、第1のメモリ13内の第2の誤り検出符号演算結果保持部132（以下、“TMP_EDC_VAL_2”と称
10 す。）に保持される。この“TMP_EDC_VAL_2”に保持される値は、前述したように、当該誤り検出回路10より得られるはずである正しいEDC演算結果と、実際に前記ステップS20の第1のEDC演算処理により得られ、現在“EDC_VAL”（第2のメモリ12）に保持されている第1の誤り検出符号演算結果との差分情報である。

15 従って、前記ステップS30の、誤り訂正回路20による誤り訂正処理において得られた前記誤りデータ位置に対して、第2のEDC処理が終了後（第1図のステップS40）、前記第2の誤り検出制御回路142は、第6図に示すように、前記“TMP_EDC_VAL_2”と、前記“EDC_VAL”のEXORをとることによって、前記“EDC_VAL”の値を更新し、正しい誤り検出符号演算結
20 果を得る（第1図のステップS50）。

そして、第2のメモリ12に保持された正しい誤り検出符号演算結果を得ると同時に、前記スクランブル除去部15にて、スクランブル成分の一括除去処理を行う（第1図のステップS60）。この処理は、前記スクランブル除去部15内に、スクランブル成分を除去するためのデータを保持するテーブルを設けておき、該
25 テーブルを用いてEDC演算処理ごのデータ符号列のスクランブル成分を一括で除去する。

ただし、前述したように、このスクランブル成分の除去処理は、第1図のステップS20のEDC演算処理の際、前記データ符号列が入力される毎にスクランブル成分を除去しながら演算処理するようにしてもよい。この場合の構成は、第

9 図に示すように誤り検出回路 10 の前段にスクランブル除去部 15 を設ければよい。また、入力データ符号列にスクランブル成分が含まれていない場合は、ECC 処理を行うために該データ符号列にスクランブル成分を付加する必要があるため、第 10 図に示すように、誤り訂正回路 20 の前段にスクランブル成分を付加するスクランブル付加部を設け、第 1 図のステップ S 10 の誤り訂正回路によるシンドローム演算処理の際、前記データ符号列が入力される毎にスクランブル成分を付加しながら演算処理するようにすればよい。

ここで、誤り検出回路 10 において、誤り検出符号演算処理を行う際に、2 つのメモリ、つまり “TMP_EDC_VAL” (第 1 のメモリ 13) 及び “EDC_VAL” (第 2 のメモリ 12) を使用する理由を以下に示す。

第 3 図に示すように、対象となるデータ符号列が連続性のない並び (第 11 図の C 2 方向) で誤り検出回路 10 に入力されるとき、1 ECC ブロックの第 0 セクタの “COL” が “1” の列における、前記誤り検出回路 10 に対するデータ入力順序は、“0” 番目、“1” 番目、“2” 番目、・・・、“11” 番目のデータまで入力された後、第 0 セクタの “208” 番目のデータではなく、第 1 セクタのデータである “12” 番目、“13” 番目、・・・のように引き続いて入力されてしまう。つまり、第 3 図に示す第 0 セクタの “208” 番目のデータが入力されるのは、第 11 図に示す 1 ECC ブロック最終セクタである第 15 セクタ最終行の “191” 番目のデータまで入力し、C 2 パリティデータである最終行の “207” 番目のデータが入力された後となる。

従って、前記誤り検出回路 10 に、第 11 図の C 2 方向にデータが入力される場合、各セクタの 1 列ごとの EDC 演算結果にしか得られないため、前回入力されたデータを元に各セクタ毎の 1 列の EDC スキップ演算を行った結果を保持するメモリが、第 0 セクタから第 15 セクタの全ての列位置 (“COL” が 1 から 42 まで) において必要となり、さらに、該各セクタ毎の各列の EDC 演算結果を、前回得られた演算結果に加算しながら保持するためのメモリが、第 0 セクタから第 15 セクタの各セクタ毎に、個別で使えるようにする必要がある。

よって、本実施の形態 1 では、各セクタ内の 1 列分の EDC 演算結果を、第 1 のメモリ 13 に保持しながら EXOR をとることによって得、1 列分の EDC 演

算結果が得られるたびに、前記第 1 のメモリ 1 3 の値を、第 2 のメモリ 1 2 内に設けられた対応するセクタ毎のレジスタに保持した後、前記第 1 のメモリ 1 3 内の値をリセットし、前記第 1 のメモリを、次のセクタの 1 列分の E D C 演算結果を得るために使用することができるようにしたものである。

- 5 次に、前述した誤り検出符号演算回路 1 1 1 における E D C 演算処理、及び誤り検出符号ステップ演算回路 1 1 2 における E D C スキップ演算処理の詳細について、第 1 5 図を参照しながら説明する。

- 10 第 1 5 図に示すように、E D C 演算回路は 3 2 ビットのシフトレジスタで構成されるため、3 2 ビット全てに 0 が存在する E D C 演算回路に対して、任意のビット値を E D C 演算回路に入力した後の E D C 演算結果である 3 2 ビットのシフトレジスタ値は、3 2 ビット中の各ビットについて、任意のビット値を入力する前の各ビット間の関係式で表現することができる。

- 15 例えば、0 の値である 1 ビットが第 1 5 図の E D C 演算回路に入力された場合、3 2 ビット中の各ビット値は、0 の値である 1 ビットが入力される前の 3 2 ビットそれぞれの状態を Bit' として以下のように示すことができる。

- Bit[0] = Bit' [31]
 Bit[1] = Bit' [0]
 Bit[2] = Bit' [1]
 Bit[3] = Bit' [2]
 20 Bit[4] = Bit' [3] ^ Bit' [31]
 Bit[5] = Bit' [4]
 .
 .
 .
 25 Bit[30] = Bit' [29]
 Bit[31] = Bit' [30] ^ Bit' [31]

前述した 1 ビットスキップ演算処理式は、前回に E D C 演算回路に対して入力されたデータと、新たに E D C 演算回路に対して入力されたデータ間の差分が 1 ビットであるときに用いる、演算回路 1 1 内の誤り検出符号演算回路 1 1 1 の演

算処理式に他ならない。

さらに、第4図に示すステップS209以外で、データの連続性をたもつために、データを168ByteスキップさせるEDCスキップ演算処理を考えると、32ビット全てに0が存在する第15図で示すEDC演算回路に対して、全
5 て0の値である1344ビット（168Byte）が入力された後のEDC演算結果である32ビットのシフトレジスタ値は、0の値である1ビットが入力される前の32ビットそれぞれの状態をBit'として以下のように示すことができる。

$$\begin{aligned} \text{Bit}[0] = & \text{Bit}'[0] \wedge \text{Bit}'[1] \wedge \text{Bit}'[2] \wedge \text{Bit}'[3] \wedge \text{Bit}'[5] \wedge \text{Bit}'[6] \wedge \text{Bit}'[7] \wedge \text{Bit}'[9] \wedge \text{Bit}'[10] \wedge \text{Bit}'[11] \wedge \text{Bit}'[13] \wedge \text{Bit}'[14] \wedge \text{Bit}'[15] \wedge \text{Bit}'[17] \wedge \text{Bit}'[18] \wedge \text{Bit}'[19] \wedge \text{Bit}'[21] \wedge \text{Bit}'[22] \wedge \text{Bit}'[23] \wedge \text{Bit}'[24] \wedge \text{Bit}'[25] \wedge \text{Bit}'[26] \wedge \text{Bit}'[27] \wedge \text{Bit}'[28] \wedge \text{Bit}'[29] \wedge \text{Bit}'[30] \wedge \text{Bit}'[31] \end{aligned}$$

$$\begin{aligned} \text{Bit}[1] = & \text{Bit}'[0] \wedge \text{Bit}'[1] \wedge \text{Bit}'[2] \wedge \text{Bit}'[3] \wedge \text{Bit}'[4] \wedge \text{Bit}'[6] \wedge \text{Bit}'[7] \wedge \text{Bit}'[8] \wedge \text{Bit}'[10] \wedge \text{Bit}'[11] \wedge \text{Bit}'[12] \wedge \text{Bit}'[14] \wedge \text{Bit}'[15] \wedge \text{Bit}'[16] \wedge \text{Bit}'[18] \wedge \text{Bit}'[19] \wedge \text{Bit}'[20] \wedge \text{Bit}'[22] \wedge \text{Bit}'[23] \wedge \text{Bit}'[24] \wedge \text{Bit}'[25] \wedge \text{Bit}'[26] \wedge \text{Bit}'[27] \wedge \text{Bit}'[28] \wedge \text{Bit}'[29] \wedge \text{Bit}'[30] \wedge \text{Bit}'[31] \end{aligned}$$

$$\begin{aligned} \text{Bit}[2] = & \text{Bit}'[0] \wedge \text{Bit}'[1] \wedge \text{Bit}'[2] \wedge \text{Bit}'[3] \wedge \text{Bit}'[4] \wedge \text{Bit}'[5] \wedge \text{Bit}'[7] \wedge \text{Bit}'[8] \wedge \text{Bit}'[9] \wedge \text{Bit}'[11] \wedge \text{Bit}'[12] \wedge \text{Bit}'[13] \wedge \text{Bit}'[15] \wedge \text{Bit}'[16] \wedge \text{Bit}'[17] \wedge \text{Bit}'[19] \wedge \text{Bit}'[20] \wedge \text{Bit}'[21] \wedge \text{Bit}'[23] \wedge \text{Bit}'[24] \wedge \text{Bit}'[25] \wedge \text{Bit}'[26] \wedge \text{Bit}'[27] \wedge \text{Bit}'[28] \wedge \text{Bit}'[29] \wedge \text{Bit}'[30] \wedge \text{Bit}'[31] \end{aligned}$$

$$\begin{aligned} \text{Bit}[3] = & \text{Bit}'[0] \wedge \text{Bit}'[1] \wedge \text{Bit}'[2] \wedge \text{Bit}'[3] \wedge \text{Bit}'[4] \wedge \text{Bit}'[5] \wedge \text{Bit}'[6] \wedge \text{Bit}'[8] \wedge \text{Bit}'[9] \wedge \text{Bit}'[10] \wedge \text{Bit}'[12] \wedge \text{Bit}'[13] \wedge \text{Bit}'[14] \wedge \text{Bit}'[16] \wedge \text{Bit}'[17] \wedge \text{Bit}'[18] \wedge \text{Bit}'[20] \wedge \text{Bit}'[21] \wedge \text{Bit}'[22] \wedge \text{Bit}'[24] \wedge \text{Bit}'[25] \wedge \text{Bit}'[26] \wedge \text{Bit}'[27] \wedge \text{Bit}'[28] \wedge \text{Bit}'[29] \wedge \text{Bit}'[30] \wedge \text{Bit}'[31] \end{aligned}$$

$$\text{Bit}[4] = \text{Bit}'[4] \wedge \text{Bit}'[24]$$

$$\text{Bit}[5] = \text{Bit}'[5] \wedge \text{Bit}'[25]$$

・
・
・

$$\text{Bit}[30] = \text{Bit}'[2] \wedge \text{Bit}'[3] \wedge \text{Bit}'[6] \wedge \text{Bit}'[7] \wedge \text{Bit}'[10] \wedge \text{Bit}'[11] \wedge \text{Bit}'[$$

[14] $\wedge \text{Bit}' [15] \wedge \text{Bit}' [18] \wedge \text{Bit}' [19] \wedge \text{Bit}' [30]$

Bit[31] = Bit' [0] \wedge Bit' [1] \wedge Bit' [2] \wedge Bit' [4] \wedge Bit' [5] \wedge Bit' [6] \wedge Bit' [8]
 \wedge Bit' [9] \wedge Bit' [10] \wedge Bit' [12] \wedge Bit' [13] \wedge Bit' [14] \wedge Bit' [16] \wedge Bit' [17]
 \wedge Bit' [18] \wedge Bit' [20] \wedge Bit' [21] \wedge Bit' [22] \wedge Bit' [23] \wedge Bit' [24] \wedge Bit' [25]
 5 \wedge Bit' [26] \wedge Bit' [27] \wedge Bit' [28] \wedge Bit' [29] \wedge Bit' [30]

前述した168 Byteスキップ演算処理式は、前回にEDC演算回路に対して入力されたデータと、新たにEDC演算回路に対して入力されたデータ間の差分が168 Byteであるときに用いる、演算回路11内の誤り検出符号スキップ演算回路112の1つの演算処理式に他ならない。そして、第4図のステップ
 10 S213においては、各“COL”値に応じたスキップ演算処理を行う必要があり、この各“COL”値に応じたスキップ演算処理式についても、前記168 Byteスキップ演算処理式と同様にして得ることができ、第4図におけるステップS213では、各“COL”値に応じたスキップ演算処理式を42通り用意すればよいこととなる。

15 なお、前述したように第4図におけるステップS213の処理のために、“COL”値に応じたスキップ演算処理式を全部で42通り用意するのではなく、前記スキップ演算を分割して処理するようにすれば、前記ステップS213におけるステップ数を削減することが可能となる。

以下、スキップ演算処理を分割処理により行う場合について、第7図の168
 20 ByteスキップさせるEDCスキップ演算処理の説明図、及び第8図の168 ByteスキップさせるEDCスキップ演算処理を分割して行う場合の説明図を用いて説明する。

第7図に示すように、第3図に示す“COL”が“0”の場合は、“ROW”が
 “11”の時点で168 ByteスキップさせるEDCスキップ演算を行う必要
 25 がある。このように、セクタ中の全てのEDC演算を行うためには、EDCスキップ演算処理は“COL”が“0”から“41”までの42通りについて用意しておく必要があり、処理資源を大きく消費してしまう。

ここで例えば、データを168 ByteスキップさせるEDCスキップ演算処理を行うことと、28 ByteスキップさせるEDCスキップ演算処理を6回行

うことが等価になる。このことを利用して、ステップS 2 1 3において、4 2通りのスキップ演算処理式を個別に用意するのではなく、例えば、データを1 6 8 ByteスキップさせるEDCスキップ演算処理を、第8図に示すように、別個に存在する、データを2 8 ByteスキップさせるEDCスキップ演算処理を6
5 回再利用する方法に変更する。なお、データを1 6 8 Byteスキップさせる場合、例えば、4 ByteスキップさせるEDCスキップ演算処理を4 2回再利用することも考えられるが、このようにすると今度は処理時間が大幅にかかってしまうという問題が生じるため、他の演算処理結果を再利用する場合には、そのスキップ演算にかかる処理時間と、用意する演算処理に必要な処理資源との兼ね合
10 いで決めるのが良い。

以上のように、本実施の形態1によれば、入力されたデータ符号列に対して、入力データ単位Byteに対応した第1のEDC演算処理を行い、その演算結果を第1のメモリ1 3内の第1の誤り検出符号演算結果保持部1 3 1に保持した後
15 に、該第1の誤り検出符号演算結果保持部1 3 1の値と、次に入力されるデータまでの差分Byte数のEDCスキップ演算処理の演算結果のEXORをとることを繰り返すようにしたので、EDC演算処理を行う際に、入力されるデータが論理的な連続性のない方向（第1 1図のC 2方向）であっても、EDC演算処理
20 を実行することが可能となり、前記ECC及びEDCの両処理の半導体集積回路の実装を実現できる。また、これにより、ECC処理とEDC処理を同時実施が可能となるので、ECCブロックをバッファから一度読み出すだけで両処理が行え、ECC処理のためにバッファから読み出した1ECCブロックを、ECC処理後に再度読み出してEDC演算処理を行う必要がなくなり、メモリバッファのバンド幅の消費、及び処理時の肥大化を低減することができる。

また、前記演算回路1 1内の誤り検出符号スキップ演算回路1 1 2における、
25 EDCスキップ演算処理を、分割処理方法を用いて行えば、セクタ内のEDC処理単位列に対応した、全てのEDCスキップ演算処理を用意することなく、別個に存在するEDCスキップ演算処理を複数回再利用することができ、この結果、処理資源の消費を抑えることが可能となる。

なお、前述の説明においては、データに論理的な連続性のないC 2方向のEC

- C処理とEDC演算処理とを同時実行する場合のEDC演算処理について詳述したが、本実施の形態1にかかる誤り検出回路10において、データに論理的な連続性があるC1方向のECC処理とEDC演算処理とを同時実行することも可能である。この場合第1のEDC演算処理は、前述したC2方向の処理とは異なり
- 5 スキップ演算をする必要がないので、演算回路11内の誤り検出符号演算回路111のみを使用し、演算処理単位（ここでは4Byte）毎に得られた演算結果を第1のメモリ13内の第1の誤り検出符号演算結果保持部131に保持していき、1セクタのEDC演算処理が終了後、前記第1の誤り検出符号演算結果保持部131に保持した演算結果を、第2のメモリ12内の対応するセクタレジスタ
- 10 に保持し、次のセクタのEDC演算処理に移行する処理を繰り返す。そして、第2のEDC演算処理は、前述したC2方向の処理と同様、前記誤り検出回路20にて得られた誤りデータ位置及び誤りデータ数値を元に、入力されるデータ符号列のうち、前記誤りデータ位置のデータに対してのみ再度EDC演算処理を行い、得られた演算結果を第1のメモリ13内の第2の誤り検出符号演算結果保持部1
- 15 32に保持し、該第2のEDC処理が終了後、その時点で第2のメモリ12に保持されている第1のEDC演算結果を、前記第2の誤り検出符号演算結果保持部132に保持された演算結果を用いて更新し、正しいEDC演算結果を得る。

産業上の利用可能性

- 20 本発明の誤り検出装置及び誤り検出方法は、ECC復号化されたデジタルデータを高倍速で記録または再生する光ディスク記録再生装置において、データを記録再生するときに、誤り訂正及び誤り検出を同時処理させる際に有用である。

請 求 の 範 囲

1. それぞれマトリクス状のデータからなる複数のセクタにより構成される対象符号列に対してシンドローム演算を行うシンドローム演算器を有し、該対象符号
5 号列に対して誤り訂正回路による誤り訂正処理を行うと同時に、該対象符号列に対して前記セクタ単位で誤り検出を行う誤り検出装置であって、
前記対象符号列の誤り検出符号を演算する誤り検出符号演算回路と、
前記対象符号列が連続性のない並びで入力された際に、該符号列の並びを連続させるようにデータをスキップさせてデータ間の連続性を補正するスキップ演算
10 を行う誤り検出符号スキップ演算回路と、
前記シンドローム演算と同じ時点で行われる、前記誤り検出符号演算回路あるいは誤り検出符号スキップ演算回路による第1の誤り検出符号演算処理を制御する第1の誤り検出制御回路と、
前記誤り訂正処理後に、該誤り訂正処理より得られた誤りデータ位置及び誤り
15 データ数値を元に、該誤りデータ位置が示すデータに対してのみ行う第2の誤り検出符号演算処理を制御すると共に、該第2の誤り検出符号演算処理による演算結果を元に、前記第1の誤り検出符号演算処理の演算結果を更新する更新処理を制御する第2の誤り検出制御回路と、
前記誤り検出符号演算回路、及び前記誤り検出符号スキップ演算回路による演
20 算結果を保持するメモリと、を備える、
ことを特徴とする誤り検出装置。
2. 請求の範囲第1項に記載の誤り検出装置において、
前記誤り検出符号スキップ演算回路は、前回までに入力された対象符号列の誤り
り検出符号を入力とし、該誤り検出符号スキップ演算回路に予め設定されたスキ
25 ップ演算を行う、
ことを特徴とする誤り検出装置。
3. 請求の範囲第1項に記載の誤り検出装置において、
前記メモリは、前記対象符号列が連続性のある並びで入力された際は、前記誤り
り検出符号演算回路、及び前記誤り検出符号スキップ演算回路によるセクタ単位

の演算結果を保持し、前記対象符号列が連続性のない並びで入力された際は、前記誤り検出符号演算回路、及び前記誤り検出符号スキップ演算回路によるセクタ毎の途中演算結果を保持する第1のメモリと、

前記第1のメモリから送信される演算結果を、前記各セクタ毎に保持する第2
5 のメモリ、と、を備える、

ことを特徴とする誤り検出装置。

4. 請求の範囲第1項に記載の誤り検出装置において、

前記第1のメモリは、前記誤り検出符号演算回路、及び前記誤り検出符号スキップ演算回路で実行される、前記第1の誤り検出符号演算処理による演算結果を
10 保持するメモリと、前記第2の誤り検出符号演算処理による、前記第1の誤り検出符号演算処理の演算結果を更新する差分演算結果を保持するメモリとを含む、
ことを特徴とする誤り検出装置。

5. 請求の範囲第1項に記載の誤り検出装置において、

前記誤り検出符号スキップ演算回路は、前記対象符号列が連続性のない並びで
15 入力されたとき、該対象符号列のうち前記セクタの非最終行においては、一定のバイト数をスキップさせるスキップ演算を行い、前記対象符号列のうち前記セクタの最終行においては、該データが存在する列位置に応じたバイト数をスキップさせる個別のスキップ演算を行う、
ことを特徴とする誤り検出装置。

20 6. 請求の範囲第1項に記載の誤り検出装置において、

前記対象符号列の該スクランブル成分を一括除去するスクランブル除去部を含む、
ことを特徴とする誤り検出装置。

7. 請求の範囲第6項に記載の誤り検出装置において、

25 前記スクランブル除去部は、前記対象符号列のスクランブル成分を除去するためのデータを保持するテーブルを有する、
ことを特徴とする誤り検出装置。

8. それぞれマトリクス状のデータからなる複数のセクタにより構成される、連続性のない並びで入力された対象符号列に対して誤り訂正単位ブロック単位で

誤り訂正を行うと同時に、該連続性のない並びで入力される対象符号列に対して前記セクタ単位で誤り検出を行う誤り検出方法であって、

前記対象符号列に対してシンドローム演算を行うシンドローム演算ステップと、

- 5 前記シンドローム演算ステップと同時に行われる、該連続性のない並びで入力される対象符号列に対して誤り検出符号演算を行う第1の誤り検出符号演算ステップと、

前記シンドローム演算ステップにおいて得られたシンドロームに基づいて、前記対象符号列の誤りデータ位置及び誤りデータ数値を計算して誤り訂正を行う誤り訂正ステップと、

- 10 前記誤り訂正ステップにおいて得られた前記誤りデータ位置及び誤りデータ数値を元に、前記対象符号列のうちの前記誤りデータ位置に対してのみ再度誤り検出符号演算を行う第2の誤り検出符号演算ステップと、

前記第2の誤り検出符号演算ステップによる演算結果を用いて、前記第1の誤り検出符号演算ステップによる演算結果を更新する更新ステップと、を含み、

- 15 前記第1の誤り検出符号演算ステップ及び第2の誤り検出演算ステップは、前記対象符号列の誤り検出符号を演算する誤り検出符号演算ステップと、前記連続性のない並びで入力された対象符号列の並びを連続させるようにデータをスキップさせてデータ間の連続性を補正するスキップ演算を行う誤り検出符号スキップ演算ステップと、を含む、

- 20 ことを特徴とする誤り検出方法。

9. 請求の範囲第8項に記載の誤り検出方法において、

前記誤り検出符号スキップ演算ステップは、前回までに入力された対象符号列の誤り検出符号を入力とし、予め設定されたスキップ演算を行う、

ことを特徴とする誤り検出方法。

- 25 10. 請求の範囲第8項に記載の誤り検出方法において、

前記誤り検出符号スキップ演算ステップは、前記対象符号列のうち前記セクタの非最終行においては、一定のバイト数をスキップさせるスキップ演算を行い、前記対象符号列のうち前記セクタの最終行においては、該データが存在する列位置に応じたバイト数をスキップさせる個別のスキップ演算を行う、

ことを特徴とする誤り検出方法。

1 1. 請求の範囲第 10 項に記載の誤り検出方法において、

前記個別のスキップ演算は、前記データが存在する列位置のうち、特定の列位置で実行されるステップ演算の演算結果を複数回利用して行う、

5 ことを特徴とする誤り検出方法。

1 2. 請求の範囲第 8 項に記載の誤り検出方法において、

前記更新ステップと同時に行われる、前記対象符号列に含まれるスクランブル成分を除去するスクランブル除去ステップを含む、

ことを特徴とする誤り検出方法。

10 1 3. 請求の範囲第 12 項に記載の誤り検出方法において、

前記スクランブル除去ステップは、1 セクタの全データが入力された後に、該セクタの全データのスクランブル成分を一度に除去する、

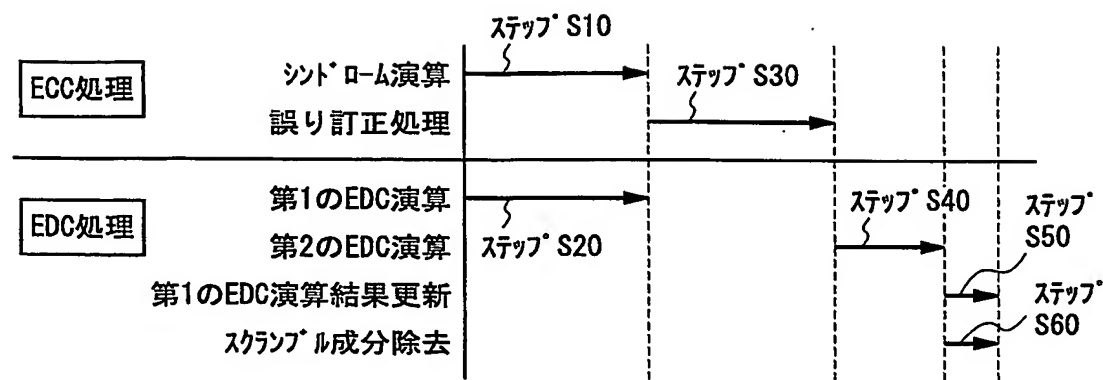
ことを特徴とする誤り検出方法。

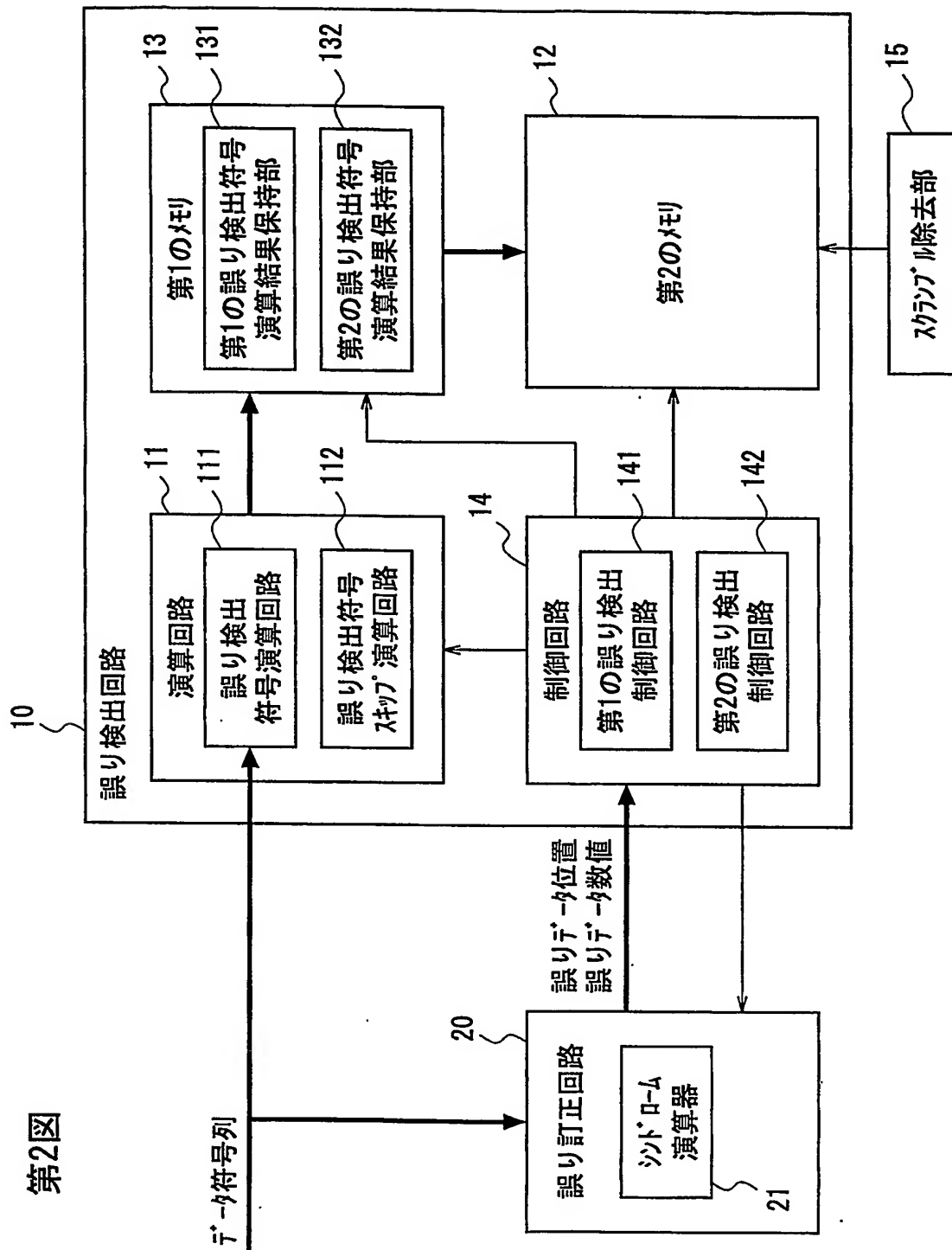
1 4. 請求の範囲第 12 項に記載の誤り検出方法において、

15 前記スクランブル除去ステップは、前記対象符号列のスクランブル成分を除去するためのデータを保持するテーブルを用いて行われる、

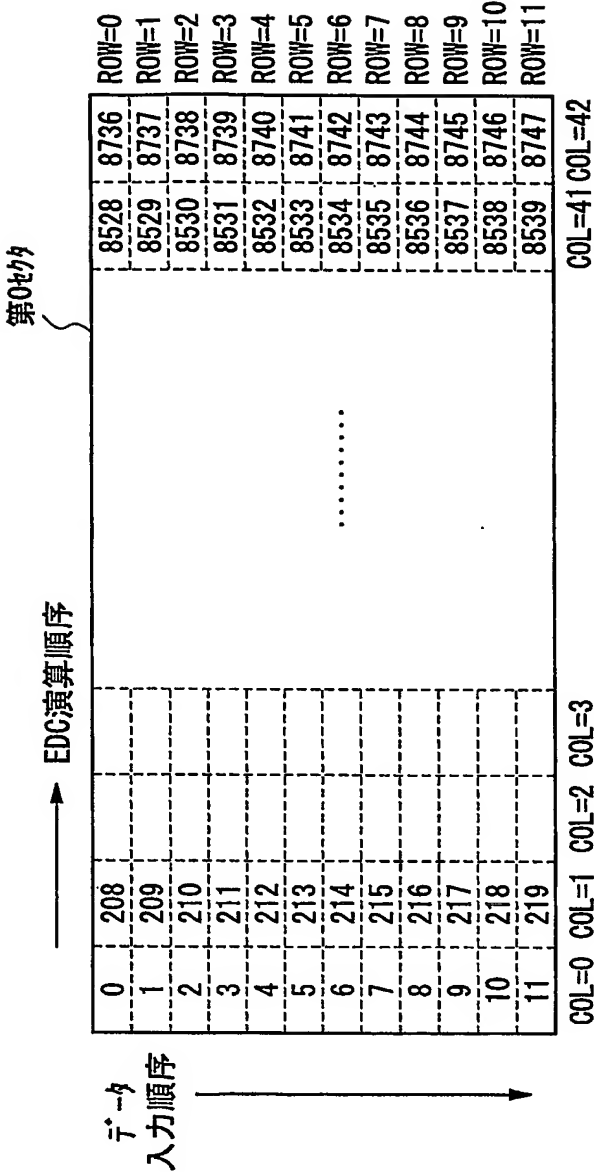
ことを特徴とする誤り検出方法。

第1図

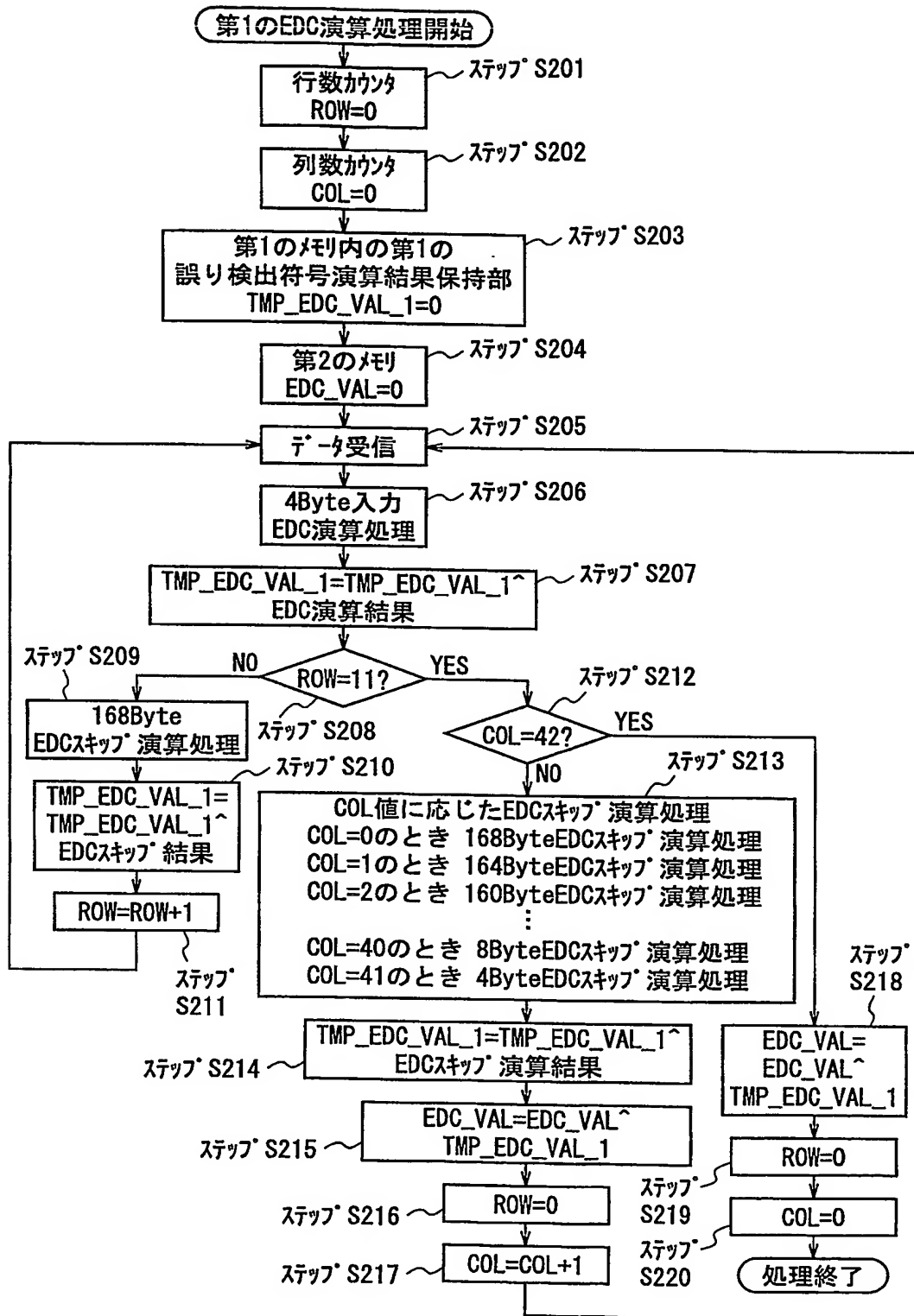




第3図



第4図



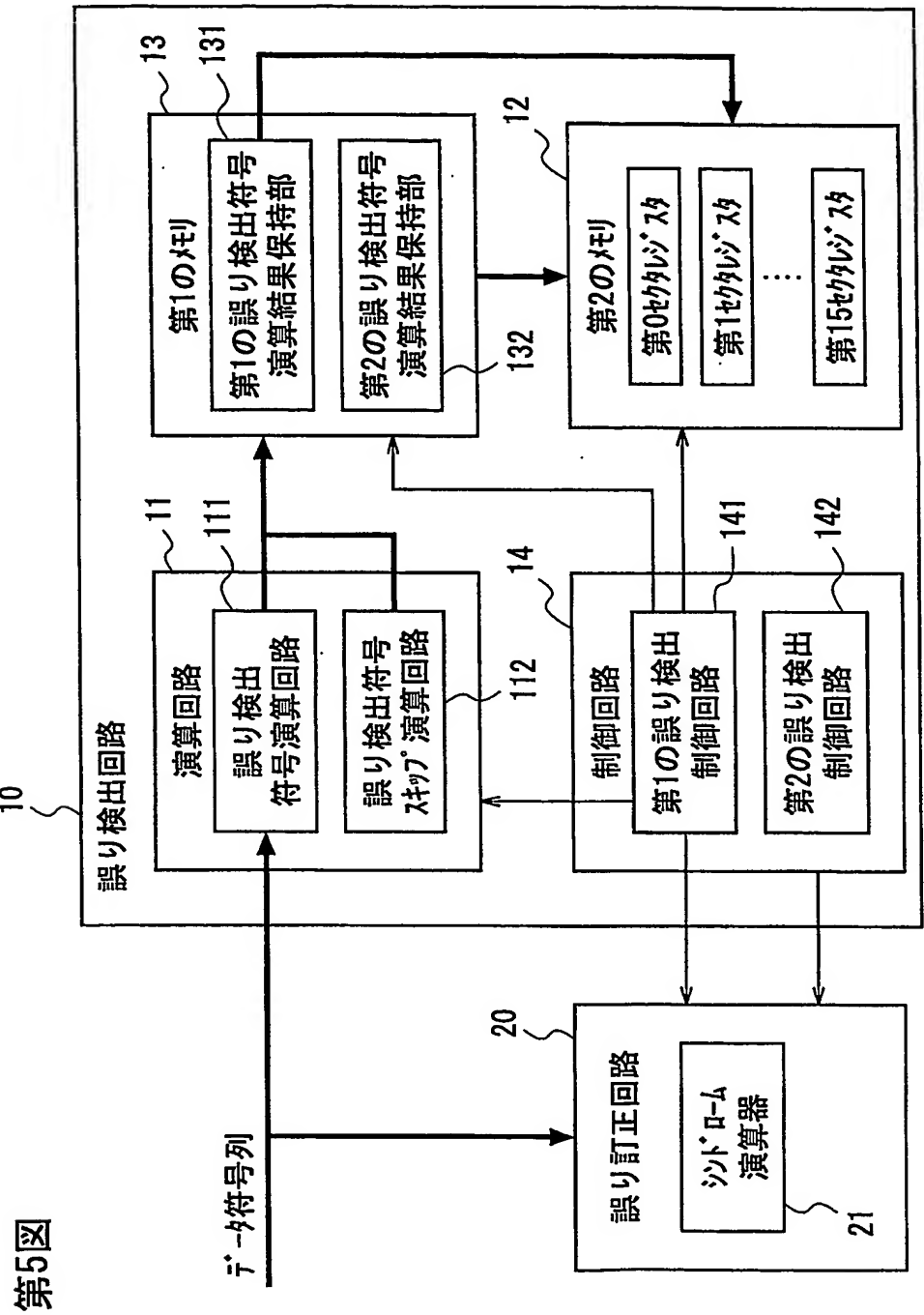
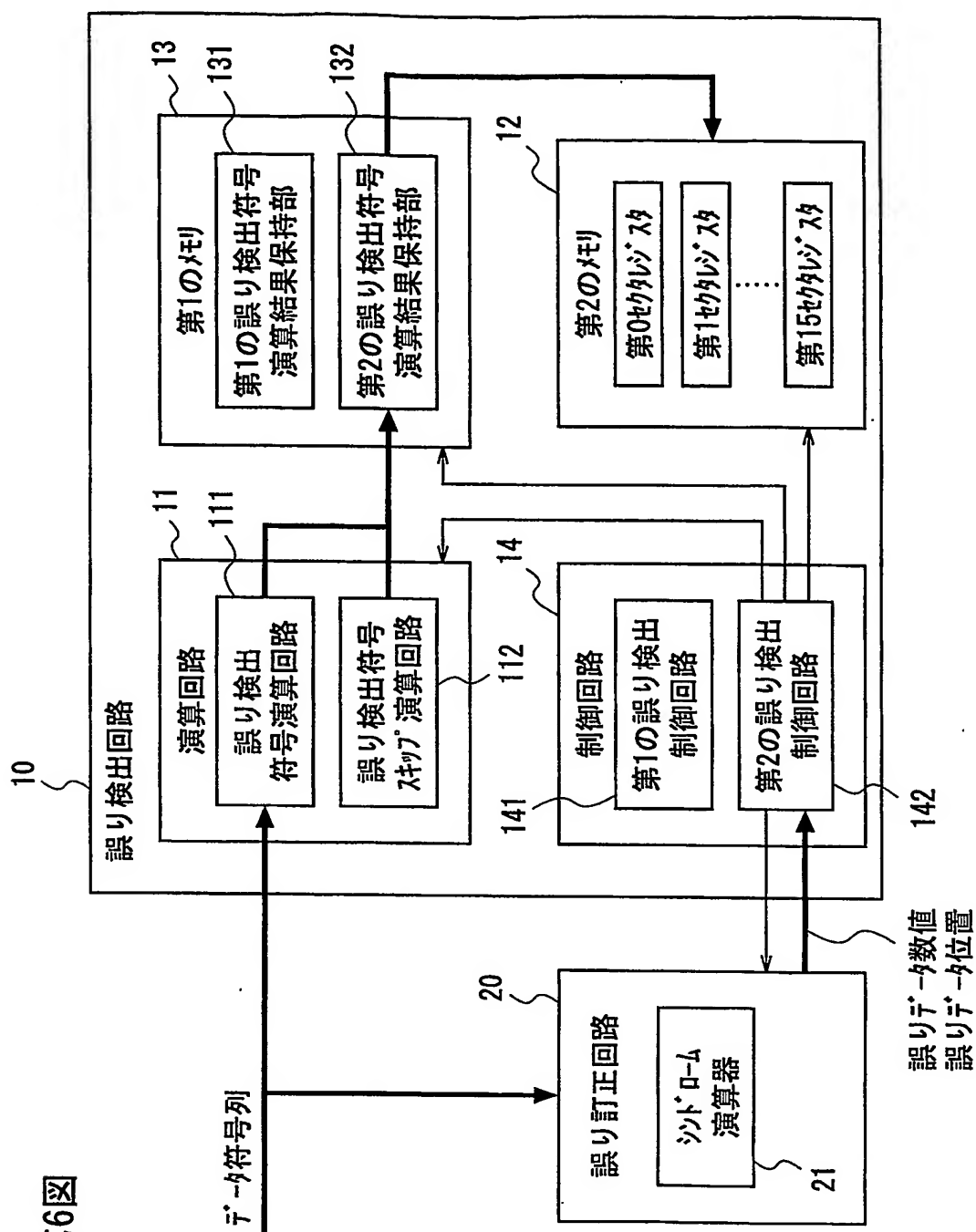
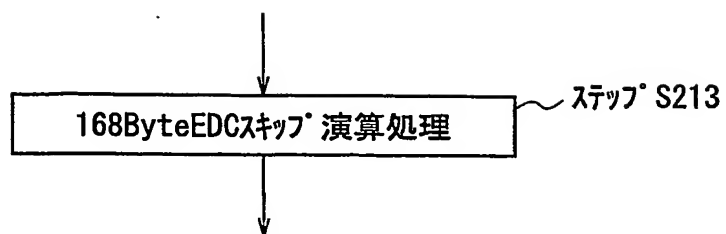


圖 6

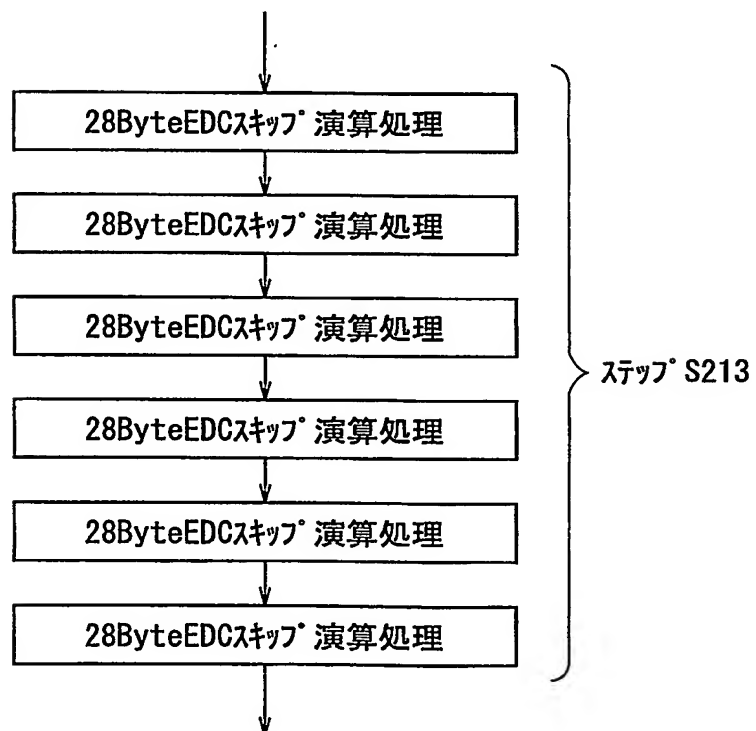


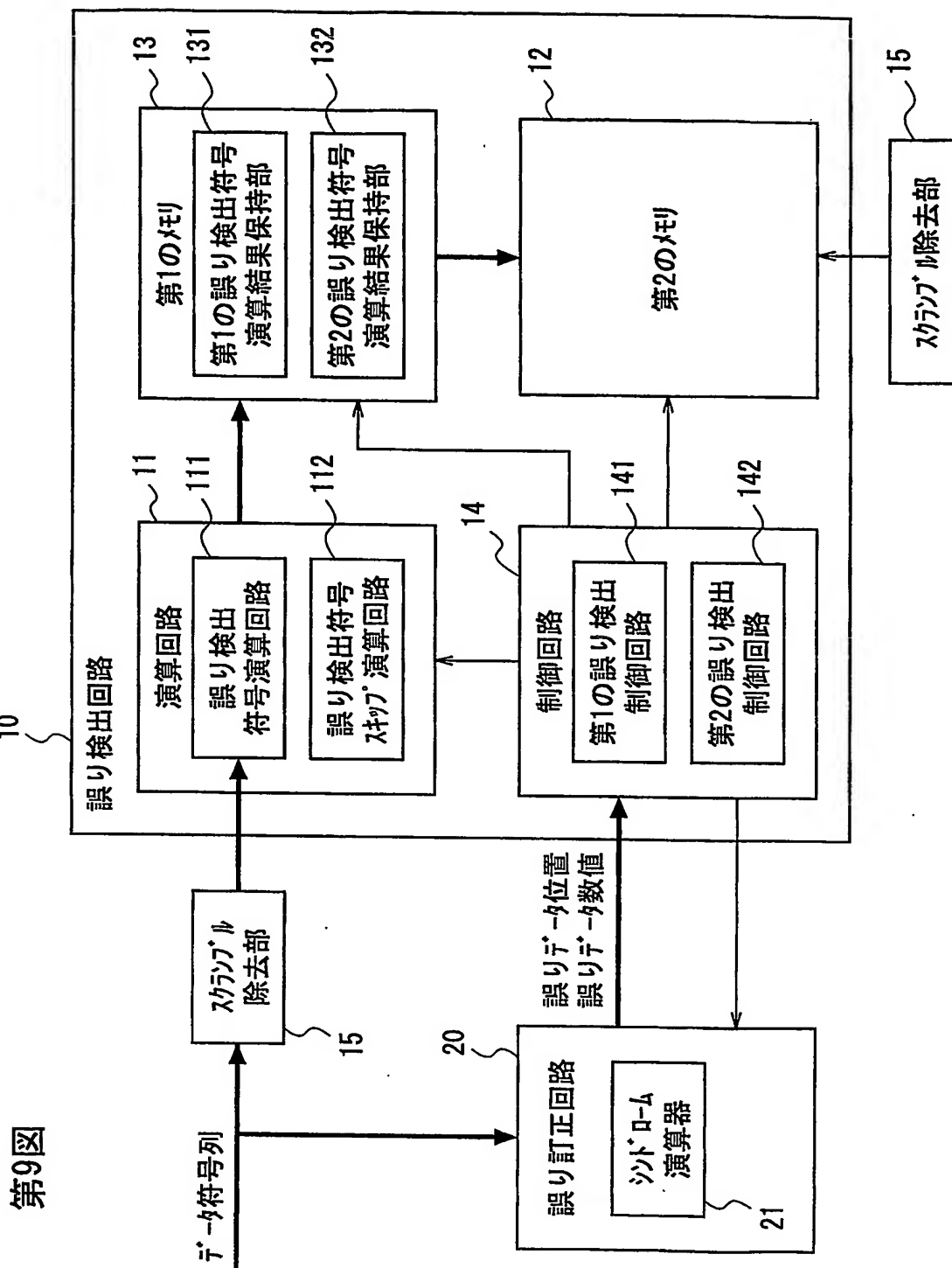
7/13

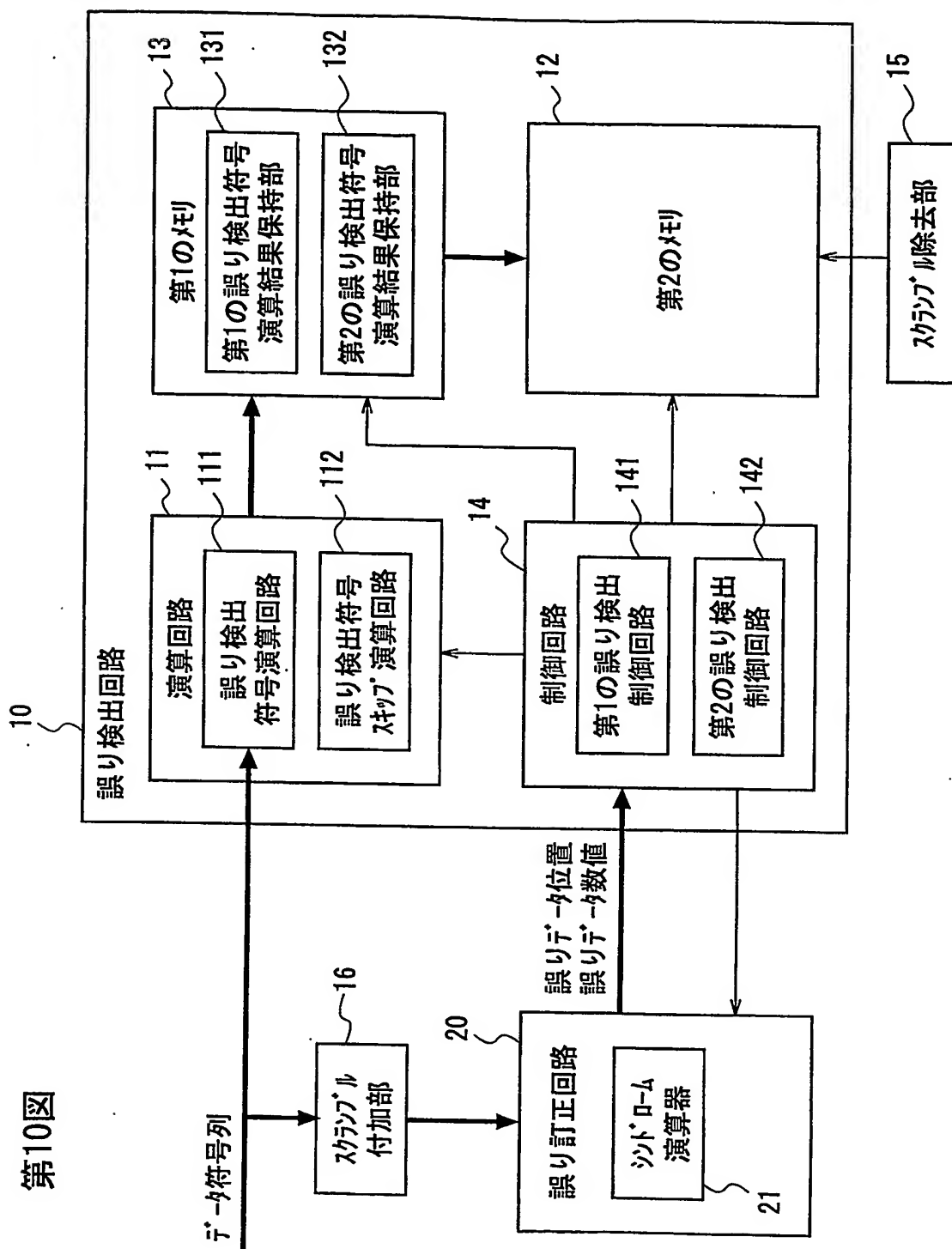
第7図



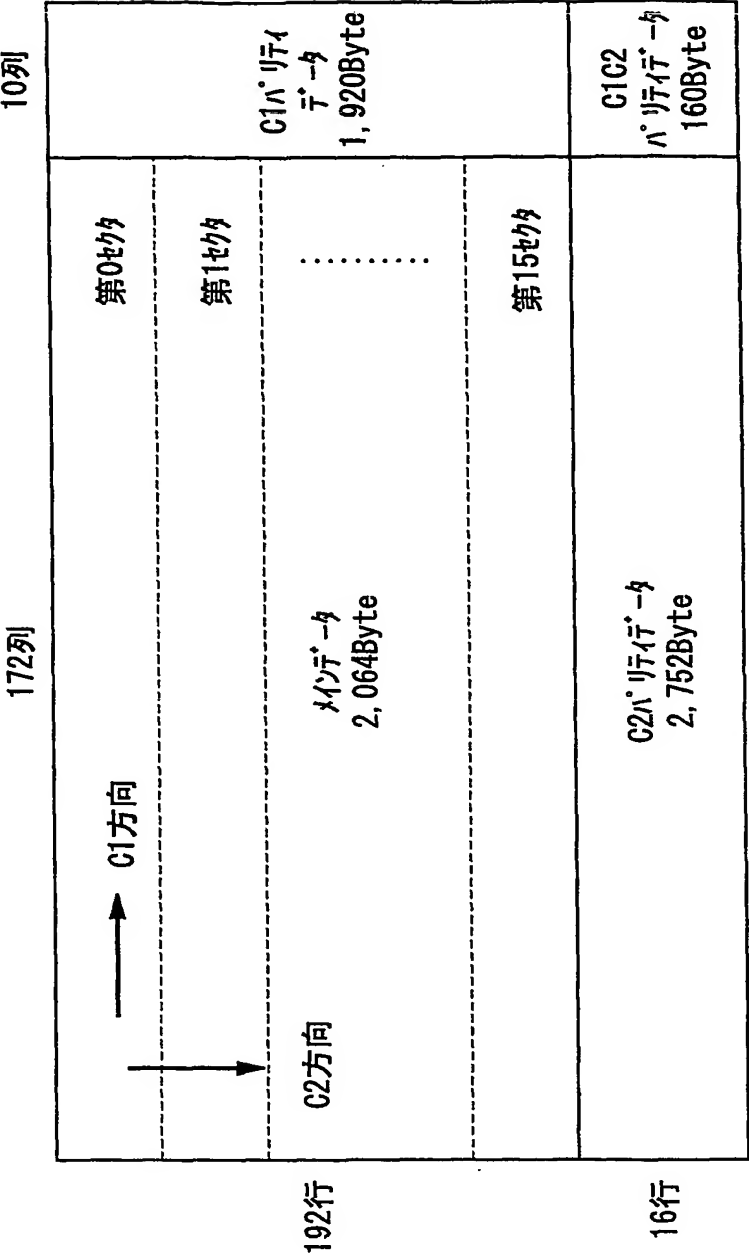
第8図



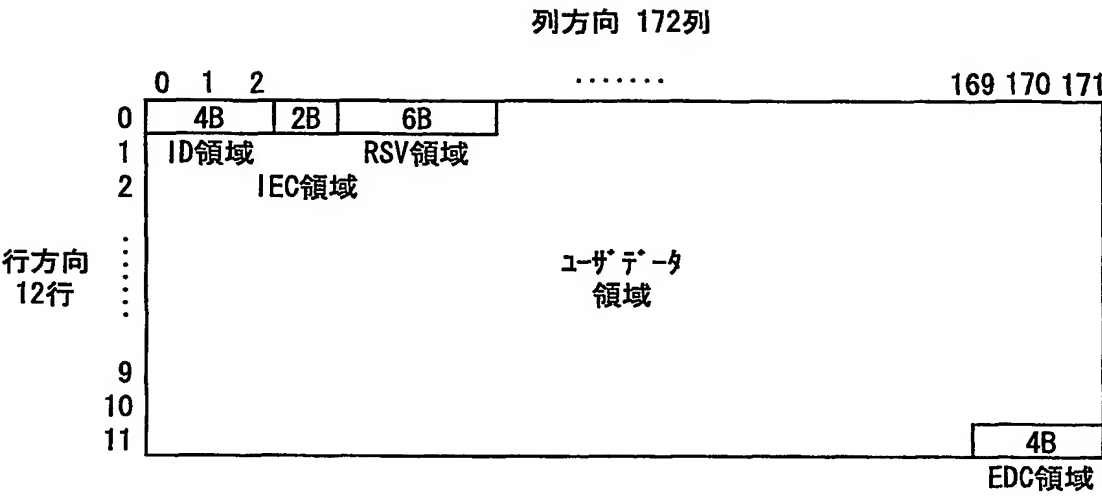




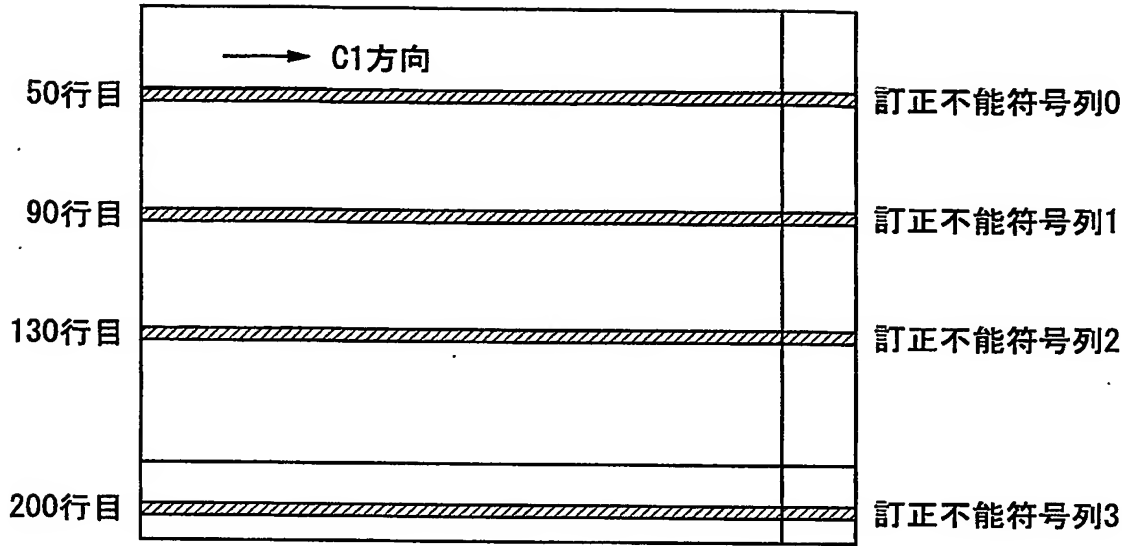
第11図



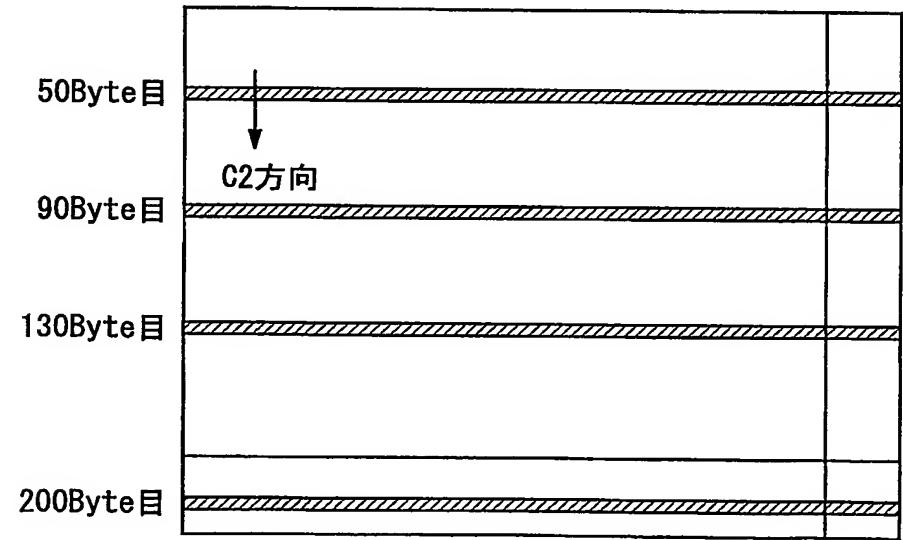
第12図



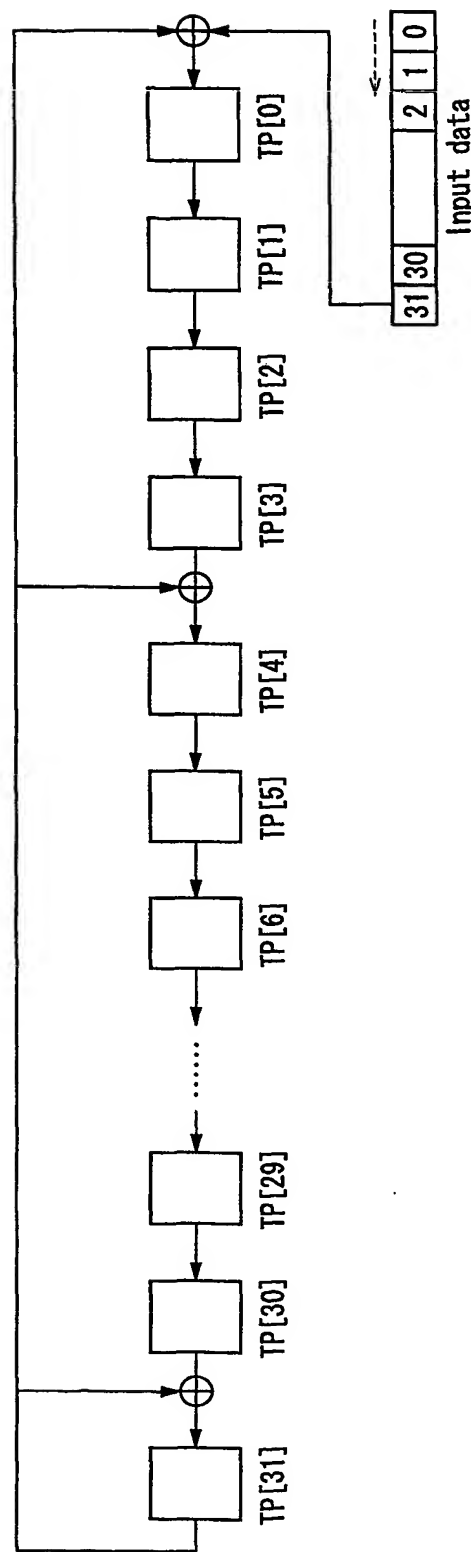
第13図



第14図



第15図



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2004/012246

A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl⁷ H03M13/15

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Int.Cl⁷ H03M13/15, 13/11, G11B20/18

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho 1922-1996 Jitsuyo Shinan Toroku Koho 1996-2004
Kokai Jitsuyo Shinan Koho 1971-2004 Toroku Jitsuyo Shinan Koho 1994-2004

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	JP 2001-23316 A (Hitachi, Ltd.), 26 January, 2001 (26.01.01), Full text; all drawings (Family: none)	1-14
Y	JP 2001-292066 A (Sanyo Electric Co., Ltd.), 19 October, 2001 (19.10.01), Par. Nos. [0001] to [0035]; Figs. 17 to 25 & US 2001/0014960 A1 & CA 1318836 A	1-14

☐ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search
19 November, 2004 (19.11.04)

Date of mailing of the international search report
07 December, 2004 (07.12.04)

Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

A. 発明の属する分野の分類 (国際特許分類 (IPC))
Int. Cl⁷ H03M 13/15

B. 調査を行った分野

調査を行った最小限資料 (国際特許分類 (IPC))

Int. Cl⁷ H03M 13/15, 13/11, G11B 20/18

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報 1922-1996年
日本国公開実用新案公報 1971-2004年
日本国実用新案登録公報 1996-2004年
日本国登録実用新案公報 1994-2004年

国際調査で使用した電子データベース (データベースの名称、調査に使用した用語)

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
Y	JP 2001-23316 A(株式会社日立製作所)2001.01.26, 全文, 全図 (ファミリーなし)	1-14
Y	JP 2001-292066 A(三洋電機株式会社)2001.10.19, 段落【0001】 - 【0035】, 図17-25 & US 2001/0014960 A1 & CA 1318836 A	1-14

☐ C欄の続きにも文献が列挙されている。

☐ . パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

「A」 特に関連のある文献ではなく、一般的技術水準を示すもの

「E」 国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの

「L」 優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献 (理由を付す)

「O」 口頭による開示、使用、展示等に言及する文献

「P」 国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

「T」 国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの

「X」 特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの

「Y」 特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの

「&」 同一パテントファミリー文献

国際調査を完了した日

19.11.2004

国際調査報告の発送日

07.12.2004

国際調査機関の名称及びあて先

日本国特許庁 (ISA/JP)

郵便番号100-8915

東京都千代田区霞が関三丁目4番3号

特許庁審査官 (権限のある職員)

藤井 浩

5K

8625

電話番号 03-3581-1101 内線 3555